US NG MODIFIED FAGAN INSPECTIONS TO CONTRO RAPID SYSTEM DEVELOPMENT

Martha Ann Griesel, Ph.D.
Linda L. Welz
Jet Propulsion laboratory
4800 Oak Grove Drive, Mail Stop 525-360
Pasadena, California 91109
Phone (818)306-6206
FAX (8 18)306-6925
agriesel@isd.jpl.nasa.gov

Abstract

The Jet Propulsion Laboratory (JPL) has been developing new approaches to software and system development to-shorten life cycle time, while maintaining product quality. One such approach has been taken by the Just-in-Time (JIT) Materiel Acquisition System Development Project. JIT is a catalog-based system for purchasing low-cost repetitively purchased commodities. The first JIT release was developed in two-thirds the traditional development time (35 weeks versus 13 months) without any decrease in product quality. To accomplish this goal the JIT project used Formal inspections (modified Fagan inspections) to focus and maintain development moment um as well as support development of high qualit y products. This paper reports on the process developed to meet the JIT development challenge successfully and on inspection, test and initial operations results.

Int roduct ion

'l'he Jet Propulsion Laboratory (JPL) has been developing new approaches to software and system development to shorten life cycle time and reduce total life-cycle cost, while maintaining product quality. One such approach has been taken by the Just-In-Time (JIT) Materiel Acquisition System Development Project. The Project is an on-going development effort to support just-in-time procurement of low-cost, repetitively purchased items. The JIT system is implemented cm three Unix servers with four special purpose PC's using Sybase as the database management system.

JIT development followed an integrated team approach with multiple deliveries, using modified Fagan Inspections to focus and maintain development momentum. This approach was adopted in response to several development challenges.

Applying Rapid System Development: To meet a schedule that was two-thirds the estimated traditional schedule without increase in budget, JIT crafted a development process based on the standard JPL development methodology and JPL experience with the Rapid Development Method [1].

Re-engineering Business Processes: JIT implemented a series of significantly re-engineered business processes. Process details were uncovered and resolved during requirements analysis and design (and a few even later), This led to a form of codesign of processes and software [2].

Implementing Radio Frequency (RF) Bar code Technology: The technical capabilities of the RF bar code equipment that were needed to increase the maintainability of the system were just becoming available. Equipment selection was difficult and final system design was delayed.

Meeting Stringent Audit Requirement The system had to meet stringent audit requirements on controlling, accessing and changing data as well as reporting all changes. These were treated as safety-related requirements [3].

Establishing Adequate Security: 'l'he authentication software had to sufficiently wrap the application and database to prevent any un-authenticated access and to prohibit unauthorized access from the application to Unix-level commands.

The JIT system also had to be sufficiently robust to support the possible addition of other commodities, some with special ordering, tracking or management requirements (e.g., microcomputer hard ware and chemicals). Some of the relevant business processes for these future commodities had not yet been re-engineered, which further complicated design decisions.

JIT Development Process

To meet these challenges, an integrated development team was formed consisting of nine individuals (system engineers, software engineer-s and programmers). Each team member participated in all aspects of system development: system engineering, implementation, test and operations for the first 60 days. There were no role statements. Team members were expected to step in and cent ribute wherever they could.

The team implemented a development process that was efficient, supported a high degree of concurrency in business process and software design and produced a high quality, tightly controlled system. The goals of the development process were: automate as much as practical, leaving the design team free to focus on doing the engineering; eliminate rework (this includes writing each piece of documentation only once); reduce communications overhead within the development team; and maintain development momentum,

Automate the process: JIT chose a requirements and design method that was familiar to the team and a supporting CASE tool. Training in the tool was provided up-front. The tool maintained Data Flow Diagrams (DFDs), Entity Relationship Diagrams (ERDs) and Data Dictionary as critical parts of the requirements and design documentation, as well as provided configuration management for both requirements and design.

Eliminate rework: The team kept pieces of the DFDs, ERDs and Data Dictionary small enough to minimize inspection rework. No documentation associated with these products was written until the engineering was stable. Even for the management plan, network schedules and related budgets were developed in a standard planning tool without being embedded in a formal written plan until the design review.

Reduce communications overhead: All team members working on a product participated in all design team meetings. Five additional team members were involved primarily in vendor procurement, business process re-engineering, budgeting and scheduling. One or more of these application specialists were involved in design meetings as needed. interaction with the initial vendor procurements was critical because the JIT system placed requirements on the vendor. In turn, the vendor choice could require additional or different capability from the JIT system.

Maintain development momentum: An alternating pattern of design team meetings and inspections, discussed in the next sect ion, kept the team focused on the engineering issues at hand, Creating a framework of inspected DFDs and ERDs avoided requirements and design churning and kept communications open.

In crafting this development process, JIT viewed inspections as a method that, combined with the chosen design method, supported rapid development of a quality system,

JIT Inspection Process

'] 'he JIT inspection process was tailored in several ways to the rapid development environment. The tailoring reduced inspection implementation time wherever possible while maintaining a quality inspect ion program, The elements that were tailored are team composition and role assignment, training, process steps and defect categorization, Each of these elements is discussed below.

JIT used inspections for requirements (DFDs) and architectural design (ERDs and Data Dictionary), Development momentum was maintained by alternating at most three, more usually two or even one, design team meetings with each inspection, Each work product underwent several inspections, The goal was to have measurably more product in the development baseline at the conclusion of each inspection cycle.

Inspection Team

The inspection team roles of Moderator, Author, Reader, Recorder and Inspector were applied to each inspection. A description of these roles is provided in Table 1. The JIT inspection team was formed from the design team. The core inspection team consisted of six individuals, with additional members who represented application specialties such as procurement and provided support as needed. An outside moderator from quality assurance was included.

Table 1- Inspection Roles

Moderator - Conducts inspection process including inspection meetings and collects inspection data. Plays key role in all stages of process except rework. Performs duties of an inspector.

Author - Provides information about work product during all stages of process. Corrects all major defects and any minor defects that cost and schedule permit. Performs duties of an inspector.

Reader - Reads or paraphrases work product in detail to guide team during inspection meeting Performs duties of an inspector.

Recorder - Accurately records each defect found during inspection meeting. Performs duties of an inspector.

Inspector - Finds defects in work product from a general point of view as well as from specific area of expertise.

Moderator - The quality assurance moderator had several years of inspection experience, both in implementation and training of inspections. While not a member of the JIT design team, the moderator did have previous development experience in large administrative systems. The combination of previous inspection and development experience as well as project independence enabled the moderator to focus on the inspection process, while understanding the technical discussions and directing the inspection team when necessary.

Since the moderator was independent of the design team, the JIT system engineer accepted several responsibilities of that position. The system engineer planned and organized the inspection meetings, performed follow-up and analyzed the inspection data.

JIT conducted a few inspection meetings without the support of the independent moderator. During these meetings another inspection team member, thus also a design team member, performed the moderator role. With this combination of roles, the development moderator had difficulty directing the meeting flow. They instead became too deeply involved in the technical discussions to adequately control the meeting. Therefore, the independent moderator participated whenever possible.

Author - JIT development during requirements and design was truly an integrated team effort. Therefore, no one person assumed the role of author. Every member of the core inspection team could be considered an author. The team could therefore have open discussions during the meetings because no one was concerned about author identity or ego. The team was able to review and resolve issues in a productive manner.

Reader - Several team members assumed the role of reader. Since each inspection team member was familiar with the products being reviewed, each was able to easily adapt to being a reader.

Recorder - The system engineer performed the role of recorder for all inspections. Recording during an inspection is a difficult job since it requires concentrating on the on-going conversation while summarizing and documenting the previously discussed defect. Having a system view as well as detailed knowledge of the developing JIT system enabled the system engineer to summarize and record defects quickly and succinctly.

Inspector - When transitioning a team intimately involved with product development to inspections, it might be assumed that no new issues would arise during the inspection meeting. However, the inspection process provided the opportunity for the "ghost" inspector to appear. The synergy of the inspection meeting brought out issues and problems that no one had identified during design team meetings.

1nspection Training

When JIT began implementation of inspections no one on the design team was formally trained in the process. Training is essential for quality inspections to ensure that inspectors understand the focus and goals of inspections as well the process. The independent moderator trained the JIT system engineer before the first inspection. At this time the independent moderator and the system engineer also determined how to implement inspections for JIT. A subset of the JPL inspection process was defined in a set of rules for JIT (Table 2).

At the first inspection meeting, real time training was provided to the newly formed inspection team. The moderator introduced the inspection team members to the core set of rules established for JIT. The moderator also described the basic inspection process, including the purpose and focus of the meetings. Training continued throughout the meeting as team members adapted to their new roles and gained experience in inspections, If the team drifted from the inspection format, the moderator stopped the meeting, refocused the team and pointed out the differences between design and inspection meetings, The team concurrently learned and implemented the inspection process, making effective use of time in a tight development schedule, With this training, the team was able to perform quality inspections and assume roles of reader, recorder, author or in specter.

Table 2- JIT Inspection Rules

Objective: To identify and record defects in the work product

Procedures:

- Limit meeting to 2 hours
- As reader presents material, revi ewers in terrupt when an issue is noted
- State issue in terms of a problem, not a solution
- Avoid discussion of solutions and design issues in the inspection meeting
- **Review team must reach consensus cm** disposition of issue
- Limit discussion of an issue 10 a few minutes (3 to 4). If no consensus is reach, classify as open issue and move on
- Record all open issues

inspection Process

JIT inspections followed the JPL inspection process [4,5], which is modeled on the process developed by Michael Fagan [6]. The JPL process includes the six basic steps of Planning, Overview, Preparation, Meeting, Rework and Follow-up. JPL has added an additional step of Third Hour [7] to Fagan's original process to provide time to resolve open issues identified in the in spect ion meeting.

Planning - The JIT system engineer conducted planning instead of the independent moderator. Since the system engineer was in daily contact with the design team, less time and effort were required to accomplish the planning tasks of coordination and facilitation, The system engineer did contact the independent moderator for advice and assistance as needed. For example, the moderator provided advice in choosing the size of the product for inspection. The system engineer could incorporate the moderator's advice with an understanding of the team's knowledge of the product and choose the correct amount of material to review at a single meeting.

Overview - The inspection overview was eliminated because the inspection team was formed from the design team, Thus every member had an in-depth knowledge of the work product to be reviewed as well as the system into which the product would be incorporated,

Preparation - JIT reduced preparation for their inspections. The core inspection team was involved in the development of the work products being inspected. Individual application experts were also involved in the design meetings prior to an inspection meeting and, therefore, required limited preparation. However, the independent moderator did perform preparation since the moderator was not familiar with the work products before inspection.

Meeting - inspection meetings were limited to two hours, a rule closely followed at JPL. This time limit is set because experience has shown that defect detection efficiency drops dramatically after two hours. As the team gained experience, JIT evolved an hour and a half limit,

Defect description and location were recorded. Defect classification (Table 3) was limited to defect severity (major, minor or open) and defect category (mi ssing, wrong or extra). This limitation on defect classification was chosen as an optimization of meeting time and need for data, The defects were classified to provide sufficient i n format i on for rework. However, the meet i ngs were not hindered by discussion of detailed defect types.

Table 3 - Defect Classificat ion

Severity

Major - An error that would cause a malfunction or prevent the attainment of an expected or specified result. An error that would result in a future approved change rcq uest or problem report.

Minor - A violation of standards, guidelines, or rule.s, which would not result in a deviation from requirements if not corrected, but could result in difficulties with operations, maintenance or future development [8].

Open - Issues that can not be resolved during the inspection meeting. Team members are assigned to resolve open issues during Third Hour.

Categories

Missing - Add information

Wrong - Correct erroneous i tem

Extra - Delete unnecessary information

The biggest challenge during inspection meetings was limiting the discussion. A previous project at JPL, faced with a similar situation, had instituted a "3 minute rule." If the inspection team could not agree that an issue was actually a defect within three minutes, the issue was recorded as an open issue and addressed during Third Hour. This rule kept the JIT meetings moving, allowing the team to cover as much material as possible, The rule was especially important if an inspection team member had time limitations and could not attend future inspection meetings,

For JIT, the independent moderator had to balance the need for discussion with the need to keep the meeting moving. If the work product was undergoing its first inspection, the moderator allowed more discussion, Team members often identified new issues in the initial inspection meeting. At later inspections for the same work product, the moderator did limit discussion, thus keeping inspections from becoming design team meetings.

Rework - Al 1 team members were product authors, therefore any one team member could perform rework. Rework was usually shared among several team members. Rework was therefore rapid] y accomplished, often within a day, and the development schedule was not impacted,

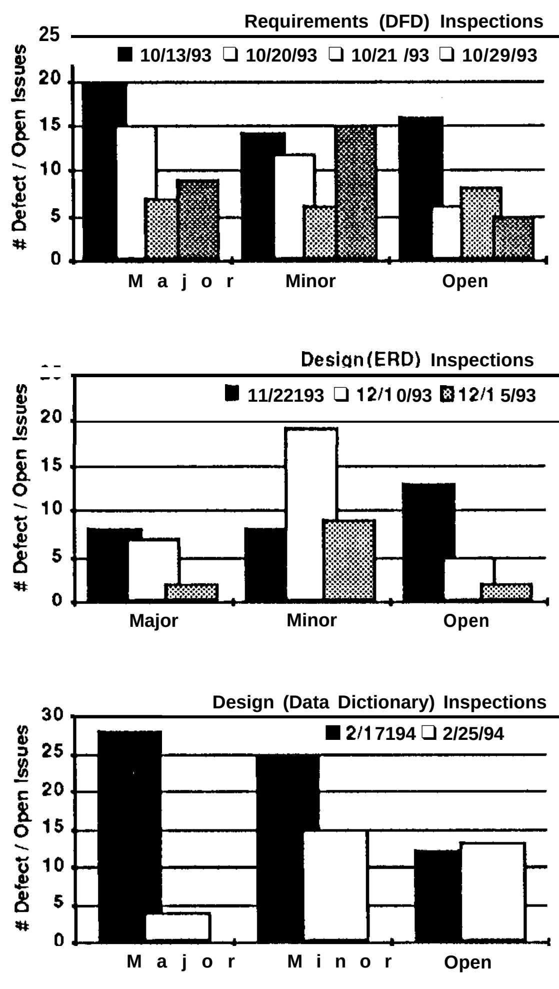
'1'bird Hour - Third hour took place in parallel to rework, Again. team members could share the work of open issue resolution and efficiently provide that information to the team members performing rework.

Follow-up - The system engineer, not the moderator, conducted follow-up. The system engineer had project knowledge that the moderator lacked and could quickly assess whether a defect had been adequately corrected. While the moderator had the technical ability to perform follow-up, it would have required more time and delayed product development, The system engineer, being co-located with the design team, also had direct access to all team members if clarification of a correction was required. Again, the moderator did not have similar access which would also have delayed follow-up, The moderator did participate with the system engineer in final inspection closure.

Inspection Results

Four requirements (DFDs) and five design (ERDs and Data Dictionary) inspections were conducted. The inspection metrics are presented in Figure 1.

Figure 1 - JIT inspection Results



Many open issues and major defects were found for the first inspection on each of the three products. Generally, these numbers decreased with each subsequent inspection. The first inspections reviewed products still under development. As the products continued to mature, fewer major defects and open issues were identified, This result indicated that the design team was able to identify and resolve issues recorded during inspections.

Minor issues generally remained the same or increased with subsequent inspections. Since the inspection team resolved major issues in early inspections, more time was available to address minor issues in later inspections.

System Test and Operations Result s

No significant defects tracing to inspected portions of requirements and design have been found in system acceptance test or operations, There were, however, several anomalies in testing that traced to areas where inspections were not used.

Anomalies fell into the following categories:

- Interfaces (vendor, order entry, JPL financial system);
- Delivery data uploads;
- Reports, which were run for the first time during system test;
- Conflicts with replication between databases when several orders were being placed concurrently.

None of the interfaces had been inspected (the order entry interface was viewed as inherited, although modified). The delivery upload code was not reviewed, The design of replication was done late and never reviewed by the entire design team.

There have been no failures in the first 33 days of operations during which 1295 items were ordered, received and delivered. Some anomalies have occurred in reporting, interfaces and replication.

Conclusions

Looking back 60 days after initial operations, the following "Lessons Learned" have been identified:

- inspections can be tailored to effectively support rapid development;
- inspections make a rapid development process a rigorous development process;
- Inspections reduce rework by eliminating requirements and design iteration;
- The addition of inspections early in the development process did not increase schedule or budget;
- Real-time training is effective for introducing inspections to a design team;
- The integrated team environment requires involvement of the entire team throughout the development process, including implementation and testing;
- Interfaces need to be inspected by the entire team;
- Inspect ions need to be used throughout the entire development process, not just requirements and design.

These lessons are being incorporated into the development of the next JIT implemental ion,

References

1. Deforrest , Lloyd R. and Dr. Lynn Graf, "Using Ada and the Rapid Development Life Cycle", Technology 2001, December, 1991.

2. Woo, Nam S., Alfred E. Dunlop and Wayne wolf, "Codesign fro m Cospecificat ion," IEEE Computer, Vol. 27, No. 1, pp. 22-47, January 1994.

3. Lutz, Robyn R., "Targeting Safety-Related Errors During Software Requirements Analysis, " SIGSOFT '93 Symposium on the Foundations of Software Engineering.

4. Kelly, John C., Joseph S. Sherif and Jonathan Hops, "An Analysis of Defect Densities Found During Software inspections", J. Systems Software, Vol. 17, pp. 111-117, 1992.

5. Sherif, Joseph S. and John C. Kelly, "Improving Software Quality Through Formal Inspections", Microelecton. Reliab., Vol.32, No, 3, pp. 423-431, 1992.

6. Fagan, M. E., "Design and Code inspections to Reduce Errors in Program Development", IBM System Journal, Vol. 15, No. 3, pp. 182-211, 1976.

7. Gilb, Tom, "The inspection process: early quality and process control" in Software Engineering Management, pp. 207 -225, 1987.

8. Buck, Frank O., "Indicators of Quality Inspections", IBM Technical Report 21.802, pp. 1-57, 1981.

Acknowledgment

The authors would like to recognized the JIT development team:

Bob Barge, Rosana Bergen, Randy Cassingham, Kay Collinson, Dennis Horgan, Rick Kleiner, Mary Malhotra, Pat Monaghan, Nilesh Patel, Audrey Ridley, Judith Skinner, Andy Thibodeaux, Bill Weisman, Dave Werntz.

Appendix: **System Description**

JIT is a catalog-based system for purchasing low-cost repetitively purchased commodities. Orders are entered on-line by Certified Users and automatically transferred to the vendor using an Electronic Data Interchange Value Added Network (EDI VAN). The vendor affixes a bar-coded shipping label, printed to JIT specifications, to each package and ships the order.

The label is scanned at the JPL receiving dock using a bar code reader that is radio-frequency (RF) linked into the JIT system. This initial scan constitutes receipt of the order and triggers the vendor payment process in the existing JPL financial system, The package is scanned again by a series of bar code readers each time it changes hands.

The JIT system tracks each event, maintaining a complete history of all processing steps from order through customer receipt and vendor payment, JIT also provides daily reports via electronic mail, showing items ordered and cost, to the customers ordering the commodities and their line and funds cognizant management. The receiving and transportation organizations receive reports on any potential problems (e.g., an order received but not delivered to the customer). Special reports are provided to catalog and contract managers as required. The system similarly supports canceling orders and returning items.

The core of the JIT system is its database, with Sybase System 10 as the database management system used during development and initial operations, The system (and the database) is distributed over three Unix servers, with each element of each table owned by the database on one server and replicated to the others that need it. Two of the servers are tightly controlled because they contain sensitive financial data. The third is an open information server that allows all employees to browse the commodity catalogs and track the status of their own orders.

Four special purpose PCs are integrated with the servers: one to communicate with the ED] VAN; one to send the electronic mail reports; one to integrate the RF bar code scanners into the system; and one to upload delivery information from hand-held bar code readers. The system communicates over the JPL network via TCP/I P and can be accessed from any PC, MAC or Unix machine on that network through an application that was written in a 4GL application development language.

The Just-in-Tinle Development Project was carried out by the Jet Propulsion Laboratory, California Institute of Technology, under contract with the National Aeronautics and Space Administration.

Fourth International Conference on Software Quality
October 3-5, 1994

Using Modified Fagan Inspections to Control Rapid System Development

Martha Ann Griesel, Ph.D. and Linda L. Welz

JPL
Jet Propulsion Laboratory,
California Institute of Technology

The Just-In-Time Development Project was carried out by the Jet Propulsion Laboratory, California Institute of Technology, under contract with the National Aeronautics and Space Administration.

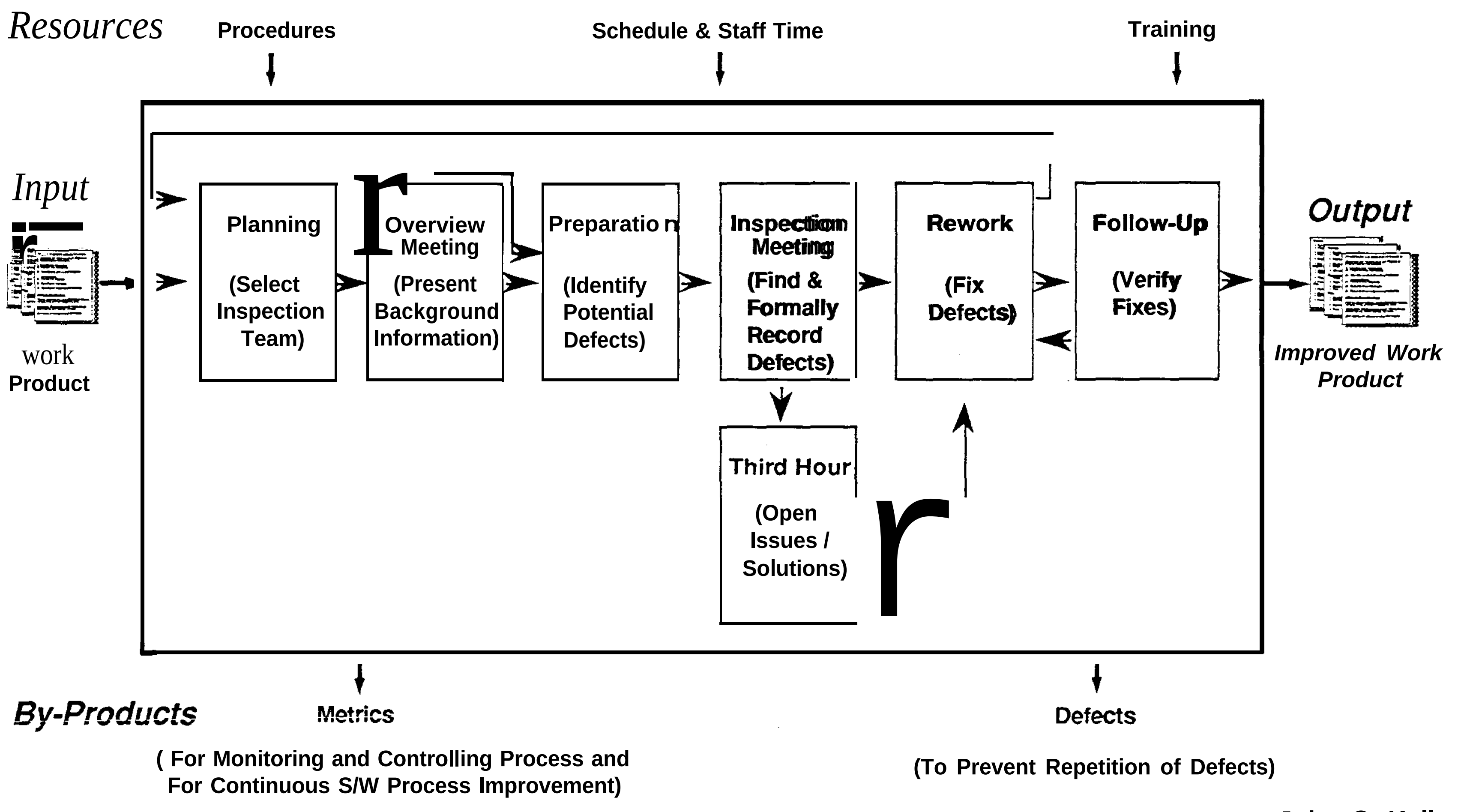
New Challenges. for Inspections

- Focus rapid system development
- Support re-engineered business process implementation
- Support total system development
  - » Radio frequency (RF) bar code technology
  - » Electronic Data Interchange
- Meet stringent audit requirements

Development Process Context

- Automate the process
  - Select a method and supporting CASE tool
  - Inspect CASE tool product
  - Use the tool to create documentation and for configuration management
- Eliminate rework
- Reduce communications overhead within the integrated development team
- Maintain development momentum

Formal Inspection Process at JPL



Resources
Procedures
Schedule & Staff Time
Training
Input
work Product
Planning (Select Inspection Team)
Overview Meeting (Present Background Information)
Preparation (Identify Potential Defects)
Inspection Meeting (Find & Formally Record Defects)
Rework (Fix Defects)
Follow-Up (Verify Fixes)
Third Hour (Open Issues / Solutions)
Output
Improved Work Product
By-Products
Metrics
(For Monitoring and Controlling Process and For Continuous S/W Process Improvement)
Defects
(To Prevent Repetition of Defects)

John C. Kelly

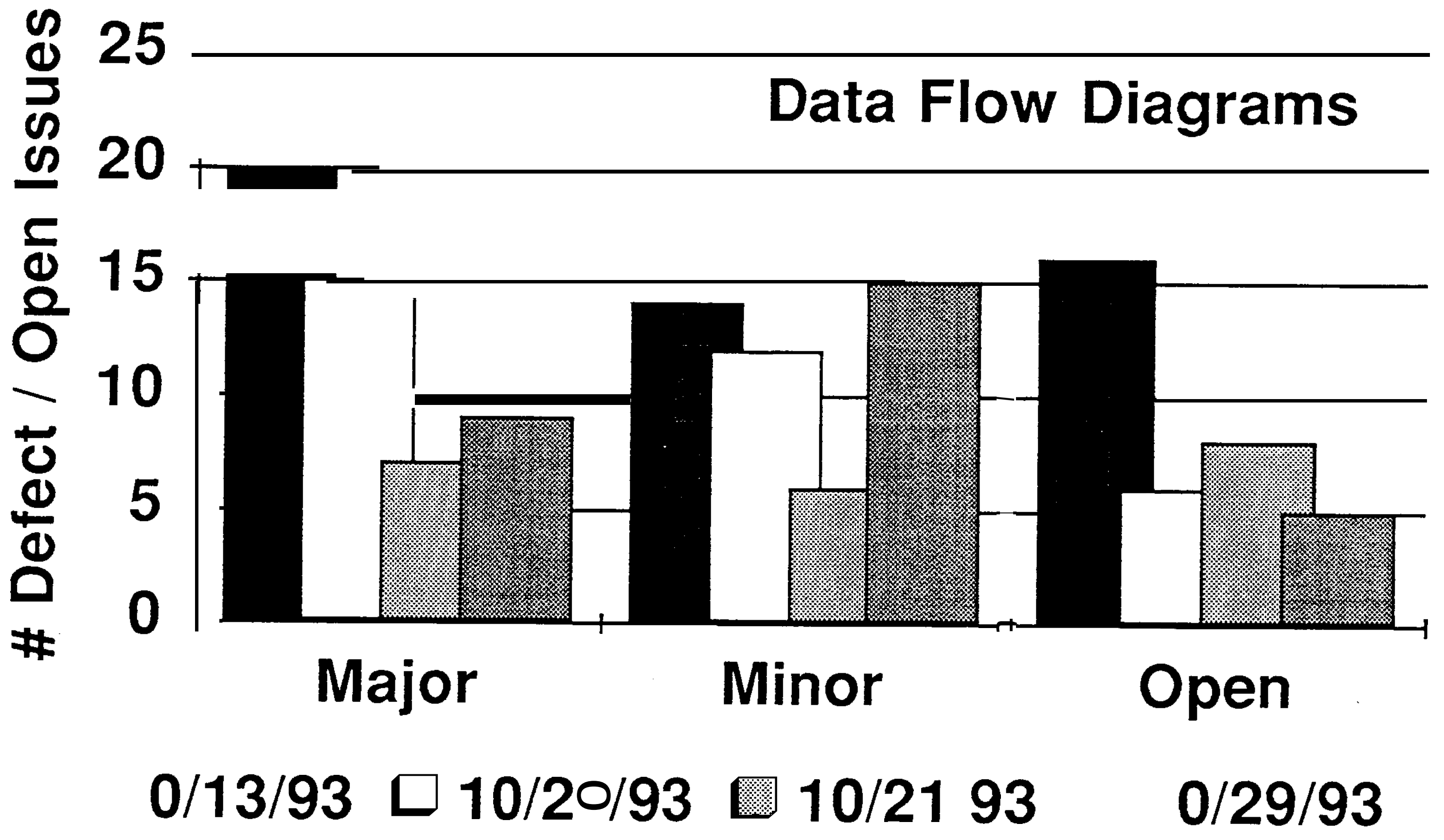
Inspection Rules for the Just-In-Time (JIT) Proiect

- Limit meeting to 2 hours
- As reader presents material, reviewers interrupt when an issues is noted
- State issue in terms of a problem, not a solution
- Avoid discussion of solutions and design issues, leave this till Third Hour
- Review team must reach consensus on disposition of issues
- Limit discussion of an issue to a few minutes. If no consensus is reached, record as open issue and move on

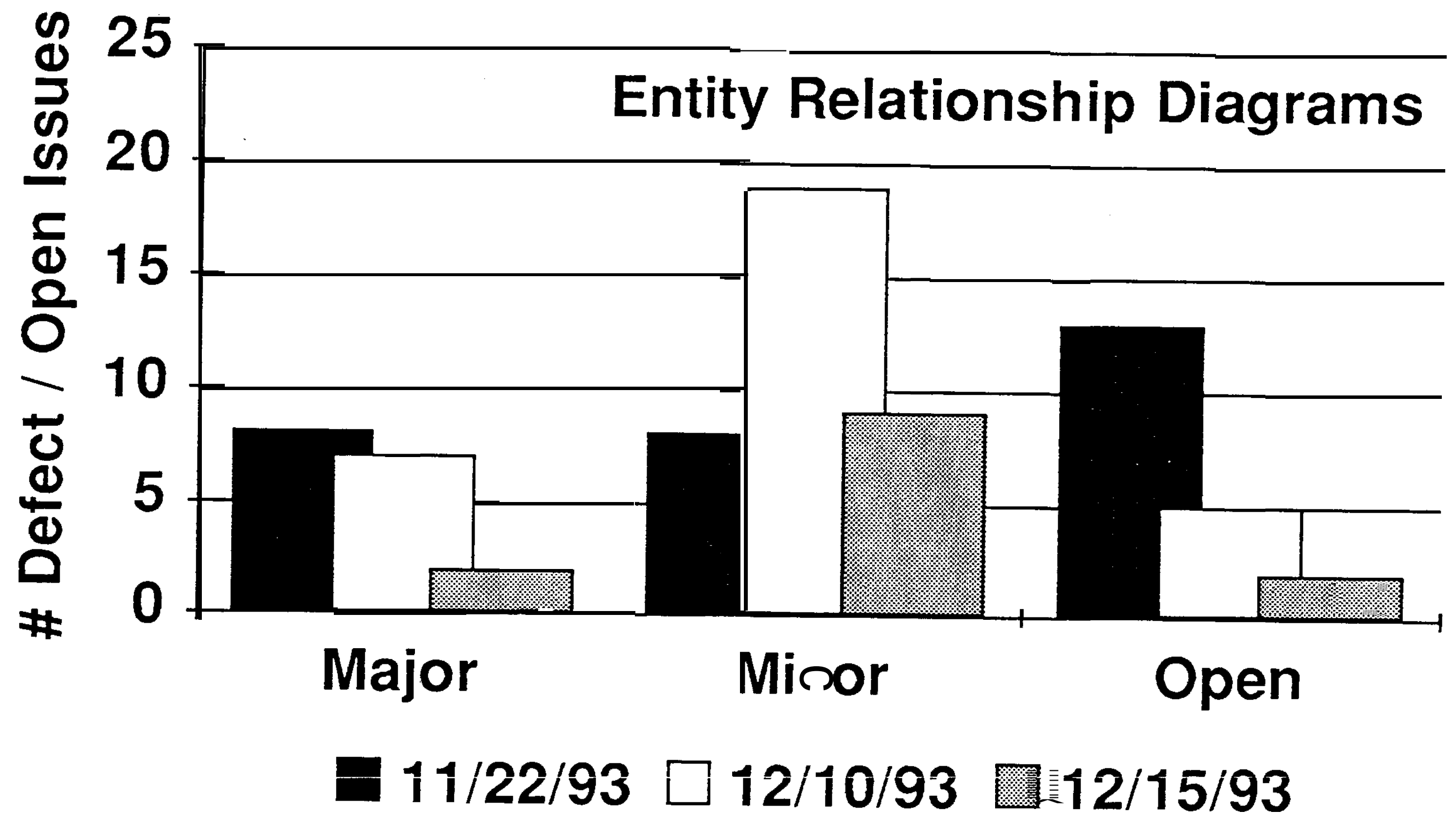
JIT Inspection Modifications

- Responsibilities of the moderator role shared between quality assurance and system engineering
- Entire development team represented author role
- Overview was eliminated
- Preparation time shortened
- Meeting discussion controlled by moderator
  - Balanced between need for review and need for design
- Rework completed by several team members
- Technical follow-up completed by system engineer

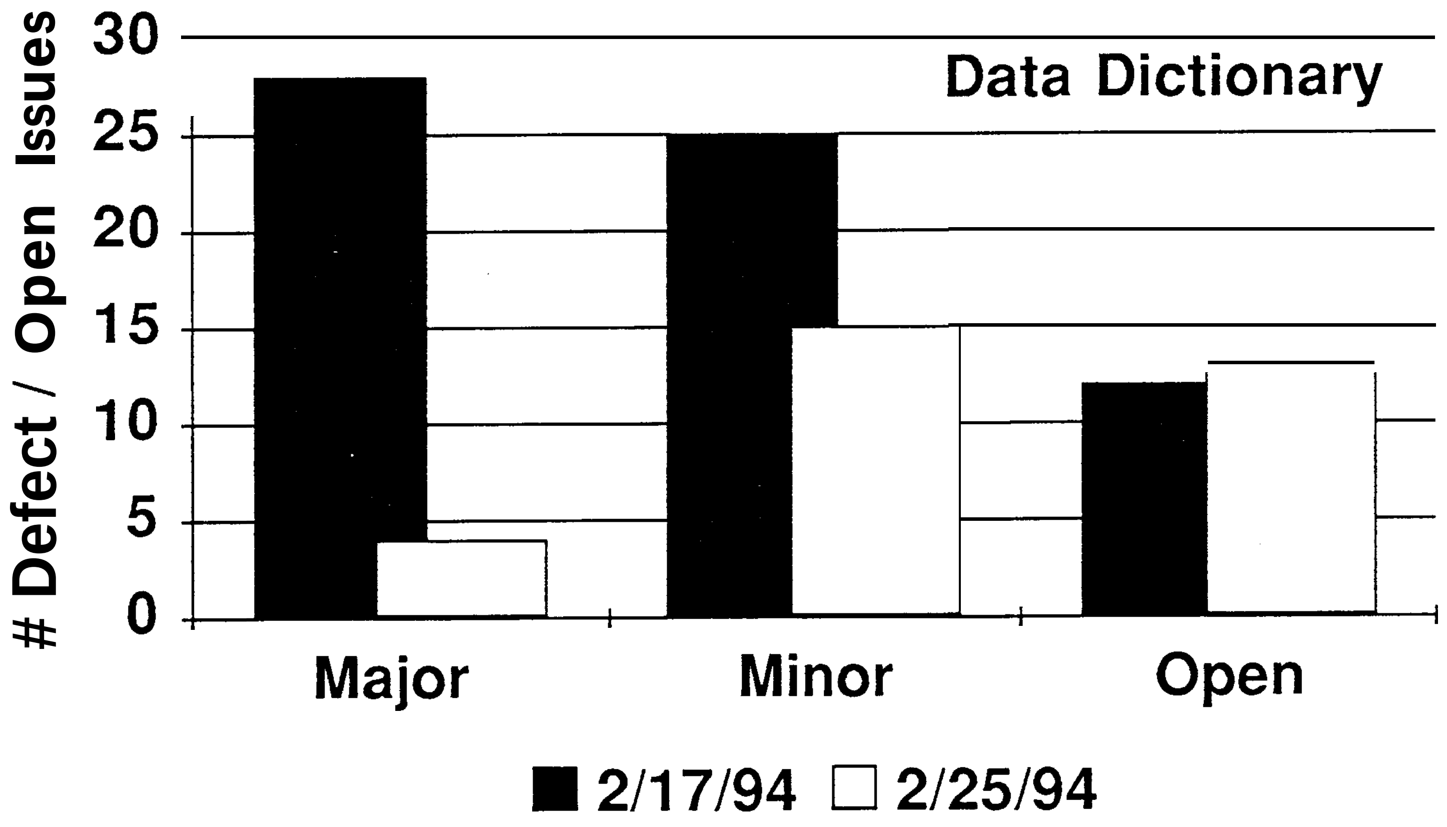
Requirements Inspection Results



Design Inspection Results



Design nspection Resu ts



Conclusions

- Inspections:
  - Can be tailored to effectively support rapid development
  - Make rapid development a rigorous development process
  - Reduce rework by eliminating requirements and design iterations
  - Did not increase schedule or budget
  - Can be introduced with real-time training
  - Need to be used throughout the entire development process